

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Bakalářská práce

2013 Václav Vaďura

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Segmentace obrazu založené na metodě
Watershed
Image Segmentation Methods Based on the
Watershed

2013 Václav Vaďura

Zadání bakalářské práce

Student:

Václav Vaďura

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Segmentace obrazu založené na metodě Watershed
Image Segmentation Methods Based on the Watershed

Zásady pro vypracování:

Segmentace obrazu bývá velmi často důležitým krokem v mnoha systémech, které mají za úkol zpracovávat a analyzovat obraz. V dnešní době existuje celá řada algoritmů právě na segmentaci obrazu. Tyto algoritmy mají za úkol rozdělit obraz na části, které spolu logicky souvisejí. Takto rozdělený obraz může být dále použit například v systémech pro rozpoznávání objektů v obraze. Jedním ze segmentačních algoritmů je i algoritmus Watershed. Tento algoritmus se vyznačuje svojí rychlostí a praktickým nasazením v oblasti medicínských snímků.

1. Seznamte se s existujícími metodami pro segmentaci obrazu.
2. V popisu se zaměřte především na přístupy založené na metodě Watershed.
3. Popište její výhody i nevýhody a možné praktické použití této metody.
4. Po nastudování a popisu jednotlivých přístupů jeden vyberte a implementujte jej.
5. V experimentální části otestujte vámi naimplementovaný přístup na vhodně zvolených snímcích.

Seznam doporučené odborné literatury:

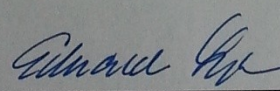
Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

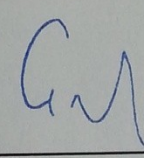
Vedoucí bakalářské práce: **Ing. Radovan Fusek**

Datum zadání: 18.11.2011

Datum odevzdání: 07.05.2013


doc. Dr. Ing. Eduard Sojka
vedoucí katedry



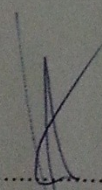

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 5.5.2013



.....

Podpis

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Radovanovi Fuskovi za odbornou pomoc a konzultaci při vytváření této práce.

Abstrakt

Tato bakalářská práce pojednává o segmentaci obrazu, která bývá velmi často používána jako jeden z kroků v systémech, které mají za úkol jak zpracovávat, tak analyzovat obraz nebo v něm vyhledat či rozpoznat objekt. Existuje celá řada algoritmů pro zpracování obrazu. My se zde zabýváme segmentačním algoritmem Watershed. Tento algoritmus se vyznačuje svojí rychlostí a například praktickým nasazením v oblasti medicínských snímků.

Klíčová slova

Segmentace obrazu, zaplavování, déšť, vodní předěl, Vincent Soille, přístupy, implementace

Abstract

This thesis deals with image segmentation, which is very often used as one of the steps in systems whose task is to process and analyze images or find or recognize the object. There are a number of algorithms for image processing. We are dealing here with Watershed segmentation algorithm. This algorithm is characterized by speed and such practical deployment of medical images.

Key words

Image segmentation, flooding, rain, watershed, Vincent Soille, approaches, implementation

Seznam tabulek

Tabulka 1: Tabulka testu 1 A).....	22
Tabulka 2: Tabulka Testu 1 B).....	23
Tabulka 3: Tabulka Testu 2.....	24

Seznam obrázků

Obrázek 1: Ukázka segmentace obrazu: a) barevné štítky/labely, b) textura, c) hrubá segmentace, d) výsledná segmentace [7].	4
Obrázek 2: Ukázka segmentace obrazu z knihovny openCV z prostředí Microsoft Visual Studia 2010: a) označení oblastí pro segmentaci, b) převedení obrázku do šedé stupnice jasu, c) zobrazené vyznačené oblasti pro začátek segmentace, d) výsledek segmentace.....	5
Obrázek 3: Hrany: skoková, šikmá, čára, střecha.	8
Obrázek 4: Průběh aktivních kontur: zleva doprava.	10
Obrázek 5: Průběh Level-Set: zleva doprava.	10
Obrázek 6: Vícedimenzionální představa: funkce Level-Set.	11
Obrázek 7: Průběh Region Growing: zleva doprava.	11
Obrázek 8: Dělení obrazu na jednotlivé kvadranty.	12
Obrázek 9: Obraz v šedé stupnici vlevo, je reprezentován jako topografický reliéf vpravo [4].	14
Obrázek 10: Nahlížení na okolní sousedy: vlevo spojitost 4, vpravo spojitost 8 [3].	15
Obrázek 11: Průběh záplavy: a) terén, b) minima v terénu, c) setkání povodí, d) watershed hráz, e) finální rozdělení do oblastí.	16
Obrázek 12: Padající kapka si zvolí cestu nejmenšího odporu.	17
Obrázek 13: Obrázek buněk, s vyznačenými počátečními body pro začátek segmentace.	19
Obrázek 14: Vlevo označené oblasti pro začátek segmentace a vpravo již výsledný obraz s uzavřenými segmenty.....	19
Obrázek 15: Zmenšený výstup Testu 1 A): a) 500 x 496, b) 250 x 248, c) 50 x 50.	23
Obrázek 16: Zmenšený výstup Testu 1 B): a) 500 x 496, b) 250 x 248, c) 50 x 50.....	24
Obrázek 17: Výstup Testu 2: a) s filtrem, b) bez filtru	25

Obsah

1	Úvod.....	3
2	Segmentace obrazu.....	4
2.1	Přístupy k segmentaci obrazu	5
2.1.1	Metody vycházející z detekce hran	5
2.1.2	Metody orientované na regiony v obraze	5
2.1.3	Statické metody	6
2.1.4	Hybridní metody.....	6
2.1.5	Znalostní metody	6
3	Detailnější pohled k vybraným přístupům.....	8
3.1	Hranová detekce	8
3.1.1	Hranové detektory	8
3.1.2	Detekce pomocí první derivace	8
3.1.3	Detekce pomocí druhé derivace	9
3.2	Houghovy transformace	9
3.3	Aktivní kontury	9
3.4	Level-set	10
3.5	Region growing	11
3.6	Split and merge.....	12
3.7	Prahování.....	12
3.7.1	Předzpracování obrazu	13
3.7.2	Automatické určování prahu	13
4	Watershed	14
4.1	Důležité pojmy	14
4.2	Záplava	15
4.2.1	Vincent-Soille algoritmus.....	17
4.3	Děšť'	17
4.4	Meyerův algoritmus	18
4.4.1	Princip	18

4.4.2	Ukázka.....	18
5	Implementace.....	20
5.1	Cíl.....	20
5.2	Implementace projektu Vincent Soille Algoritmus.....	20
5.2.1	Obsah projektu	20
5.2.2	Průběh segmentace	21
5.3	Výsledky.....	22
5.3.1	Test 1 A).....	22
5.3.2	Test 1 B).....	23
5.3.3	Test 2	24
6	Závěr.....	26

1 Úvod

Rozvoj počítačových technologií a digitální techniky neustále pokračuje a proto považujeme zpracování obrazu za velmi důležitý krok. Obraz je potřeba zpracovávat například v medicíně, důlním průmyslu, bezpečnostních systémech, fotografiích, videích.

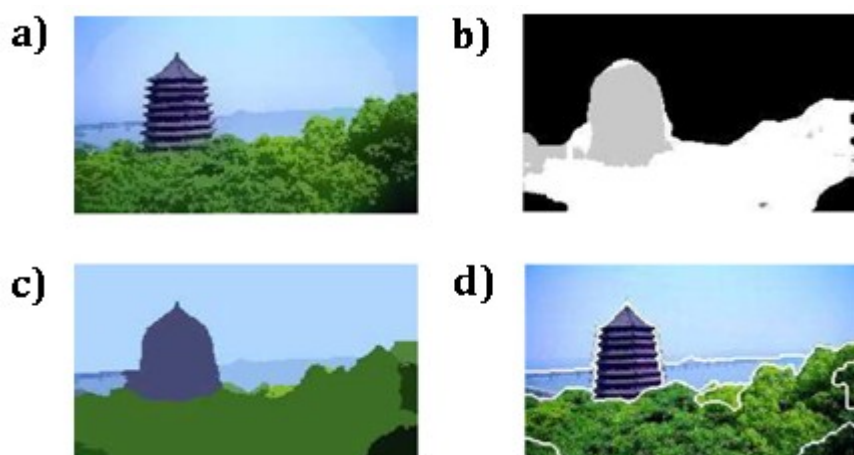
Jeden z velmi důležitých kroků při zpracování obrazu je segmentace obrazu. Segmentace obrazu má za úkol izolovat jednotlivé objekty v obraze od jeho pozadí a výsledky můžeme dále zpracovávat podle potřeby. V dnešní době existuje několik různých přístupů k segmentaci obrazu počínaje méně složitými algoritmy a složitými konče.

V této bakalářské práci budou zmíněny vybrané přístupy k segmentacím. Poté se zaměřuji na algoritmus zvaný Watershed. Konkrétně na Vincent-Soille algoritmus, který jsem si vybral k implementaci. Tento algoritmus se vyznačuje svojí rychlostí a dokáže obraz rozdělit do jednotlivých oblastí, se společnými vlastnostmi. Výsledkem bude tedy rosegmentovaný obraz, se zvýrazněnou maticí postavených hrází neboli watershedů. U této segmentace budu měřit také čas, za který segmentace proběhne.

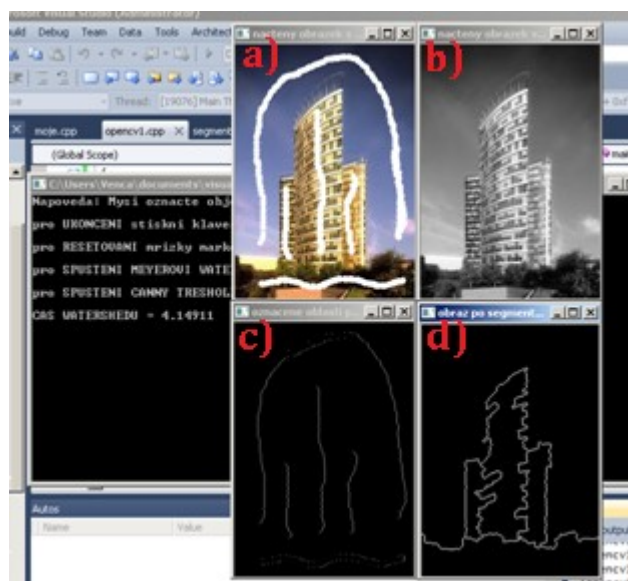
2 Segmentace obrazu

Významem segmentace obrazu je proces, který v daném obraze odděluje objekty od pozadí. Obraz je složen z pixelů a každému pixelu je přiřazen index segmentu, který vyjadřuje daný objekt v obraze. Jedná se o jeden z nejdůležitějších kroků ve zpracování obrazu. Výsledkem segmentace je tedy například detekování konkrétního objektu v obraze nebo detekování více objektů s podobnými vlastnostmi [1]. Jak taková segmentace obrazu může vypadat, je vidět na obrázku 1 a 2.

V dnešní době se můžeme setkat s mnoha segmentačními algoritmy. K segmentaci obrazu můžeme přistupovat hned několika způsoby. V této kapitole se zaměřuji na rozdělení segmentačních algoritmů a vycházím z [8].



Obrázek 1: Ukázka segmentace obrazu: a) barevné štítky/labely, b) textura, c) hrubá segmentace, d) výsledná segmentace [7].



Obrázek 2: Ukázka segmentace obrazu z knihovny openCV z prostředí Microsoft Visual Studio 2010: a) označení oblastí pro segmentaci, b) převedení obrázku do šedé stupnice jasu, c) zobrazené vyznačené oblasti pro začátek segmentace, d) výsledek segmentace.

2.1 Přístupy k segmentaci obrazu

2.1.1 Metody vycházející z detekce hran

Tyto metody jsou orientované na detekci významných hran v obraze, kdy na základě rozdílu hodnot okolních pixelů detekujeme lokální hrany. Lokální hrany detekujeme pomocí takzvaných hranových detektorů. Hranový detektor je algoritmus, který vytváří množinu hran. Touto množinou rozumíme sled bodů, pixelů nebo fragmentů v obraze.

- Hranové detektory
- Houghovy transformace
- Aktivní kontury
- Level-set

2.1.2 Metody orientované na regiony v obraze

Jde v podstatě o podobnou metodu jako je detekce hran s tím, že místo hran detekujeme v obraze regiony. Obrisy regionů nemusí být ucelené. Může se stát, že budou jistým způsobem narušené a region pak nemusí být označen jako celek.

- Region growing
- Split and merge

2.1.3 Statické metody

Základem statických metod je statická analýza obrazových dat. Nejčastěji se zabýváme hodnotou jednotlivých pixelů v obraze.

- Prahování
- Adaptivní prahování
- Connected component labeling
- Amplitudová projekce
- Shluková analýza
- Kohonenovy mapy
- Fuzzy Connectedness
- MRF (Markov Random Fields)

2.1.4 Hybridní metody

S hybridními metodami se můžeme někdy setkat i jako se sekvenčními. Některé segmentační metody je těžké zařadit do jedné z výše zmíněných skupin, jelikož mohou obsahovat z každé metody něco jiného. Mezi hybridní metody se řadí taktéž metody, které staví na matematické morfologii. Tyto metody využívají taktéž nejčastěji matematické charakteristiky obrazu, jako je například gradient, využití grafů, mřížek a podobně.

- Watershed Transform
- Neuronové sítě

2.1.5 Znalostní metody

Znalostní metody se zabývají již systémem jakýchsi předloh či modelů objektů, které hodláme segmentovat. Předlohy jsou buď generovány automaticky ze souboru trénovacích dat nebo jsou ručně vloženy na základě lidských zkušeností. V průběhu této segmentace pak algoritmus hledá transformaci již známých objektů v obraze. Toto nám umožní například definovat metodě objekt obličeje, který se skládá z očí, uší, nosu, úst, jelikož tato metoda už bude vědět, jak tyto prvky obličeje vypadají. Z toho

vyplývá, že vlastnost předem znát segmentovaný objekt na základě jeho tvaru, barvy, struktury a podobně, nám značně ulehčuje práci.

- AAM (Active Appearance Models)

3 Detailnější pohled k vybraným přístupům

V předchozí kapitole jsme se zabývali seznámením jednotlivých metod pro segmentaci obrazu. V této kapitole se seznámíme s jednotlivými vybranými technikami podle rozdělení z předchozí kapitoly. V této kapitole vycházím z [8].

3.1 Hranová detekce

Detekce hran slouží k nalezení oblastí pixelů, ve kterých se podstatně mění jas. Základním prvkem je samotná hrana. V obraze je hrana vlastnost obrazového bodu započteného jako funkce obrazu v okolí toho bodu, kde se prudce mění jas. Právě s pomocí změny jasu pixelů v obraze dokážeme rozmístit jednotlivé objekty v obraze. Existují lokální změny jasu a globální. Lokální změny jasu se nazývají jasové hrany, globální změny se nazývají jasové hraniční segmenty.

Hrany znázorňující změnu jasu dělíme na: skokové, šikmé, čáry a střechy. Jednotlivé hrany jsou znázorněny na obrázku 3.



Obrázek 3: Hrany: skoková, šikmá, čára, střecha.

3.1.1 Hranové detektory

Metody, které detekují hrany v obraze dělíme na dvě skupiny:

- Metody využívající první derivaci
- Metody využívající druhou derivaci

3.1.2 Detekce pomocí první derivace

V místě kde se nachází hrana dojde k největší změně intenzity. V homogenních oblastech je změna nulová, čili i první derivace je taktéž nula.

Princip je takový, že vypočteme první derivaci pro řádky a sloupce zvlášť. Pixely bereme tak, jak jdou zleva doprava a shora dolů.

Pro výpočet jednotlivých složek gradientu se používají hranové operátory (Sobelův, Prewittův, Robinsonův, Kirschův).

3.1.3 Detekce pomocí druhé derivace

Tam kde nepotřebujeme znát směr ani velikost hran a známe pouze místo, kde se dané hrany nacházejí používáme detekci dle druhé derivace.

V místech, kde se objevuje největší jasová změna intenzity, z čehož plyne, že první derivace bude největší, pak druhá derivace prochází přímo nulou.

Pokud chceme nalézt hrany, musíme nejprve vypočítat druhou derivaci, a až poté můžeme najít místa, ve kterých došlo k průchodu nulou.

Při řešení těchto výpočetních problémů jistě narazíme na některé záludnosti. Proto se při výpočtech používají takzvané hranové operátory. Jedny z oblíbených jsou i Laplaceovy operátory. Tyto operátory kladou důraz na středové body a na součet všech jejich prvků, který je roven nule.

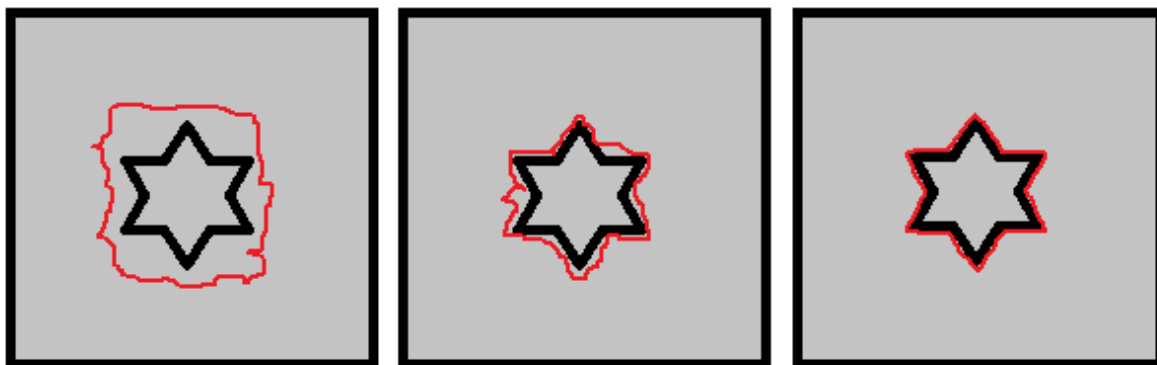
3.2 Houghovy transformace

Houghova transformace se používá pro detekci čar a rovinných křivek v obraze (jde tedy o jednoduché objekty). Tato segmentace není až tak citlivá na šum a není citlivá na porušení hranic objektů, což je její hlavní výhodou. Můžeme ji použít i tam, kde dochází k částečnému zakrytí objektů.

Mezi nevýhody této transformace patří například tlustá hrana, která se musí skládat z více přímk. Další nevýhody jsou takové, že neznáme konec ani počátek křivek, jelikož získáváme přímky a ne úsečky, rychlost, pracnost (vícenásobné vnořené cykly)- snaha o snížení výpočetní náročnosti.

3.3 Aktivní kontury

Technika aktivních kontur postupně tvaruje konturu až ke hraně objektu. Tato kontura se vlivem obrazových, vnitřních a vnějších energií tvaruje kolem objektu v obraze. Vnitřní energie kontrolují hladkost průběhu (ohyb, zlom). Obrazové energie směřují konturu směrem ke hraně objektu. Vnější energie jsou pak výsledkem počátku, kde se kontura umístí. Průběh aktivních kontur je znázorněn na obrázku 4.



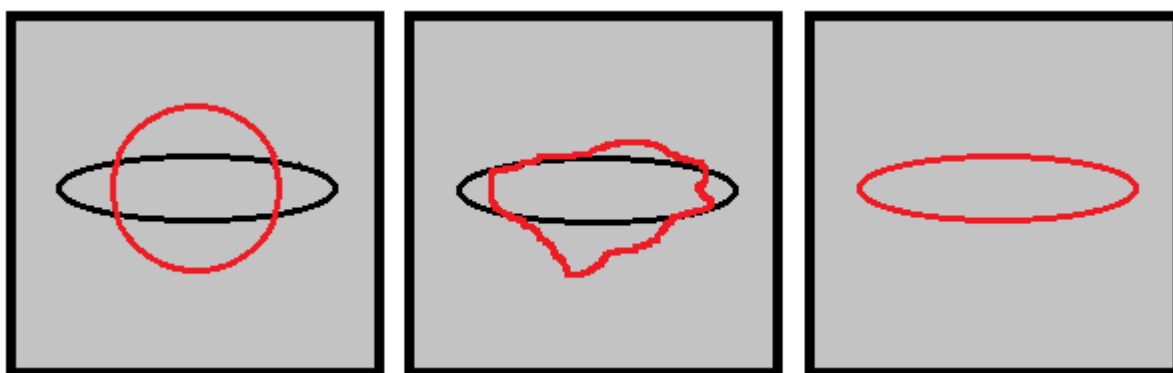
Obrázek 4: Průběh aktivních kontur: zleva doprava.

3.4 Level-set

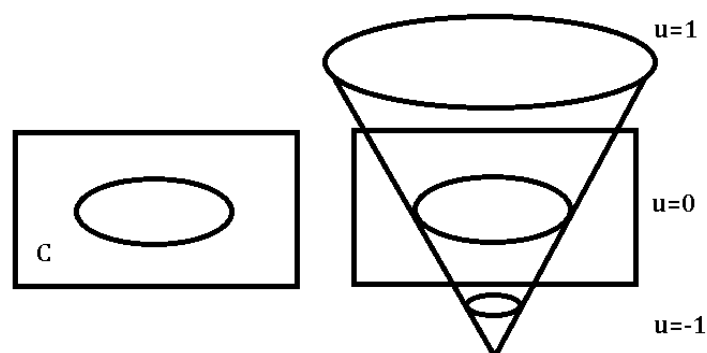
Level-set využívá obdobný přístup jako aktivní kontury. Oproti aktivním konturám se nemění tvar křivky přímo, ale pomocí funkce level-set.

Level-set je vícedimenzionální funkce. Můžeme si ji představit jako jehlan, kdy vedeme řez nulovou hladinou. Tento řez je veden v rovině xy a definuje počáteční křivku. Ke každému bodu roviny xy je přiřazena výška u nad nebo pod nulovou hladinou, jak je ukázáno na obrázku 6. Povrch funkce se pak postupně adaptuje vzhledem k zadaným metrikám křivosti, a obrazovým gradientům. Průběh Level-Set je znázorněn na obrázku 5.

Využití Level-Set je nejefektivnější pro složitější tvary v obraze. Inicializace probíhá manuálně a přibližně odpovídá cílovému tvaru křivky.



Obrázek 5: Průběh Level-Set: zleva doprava.

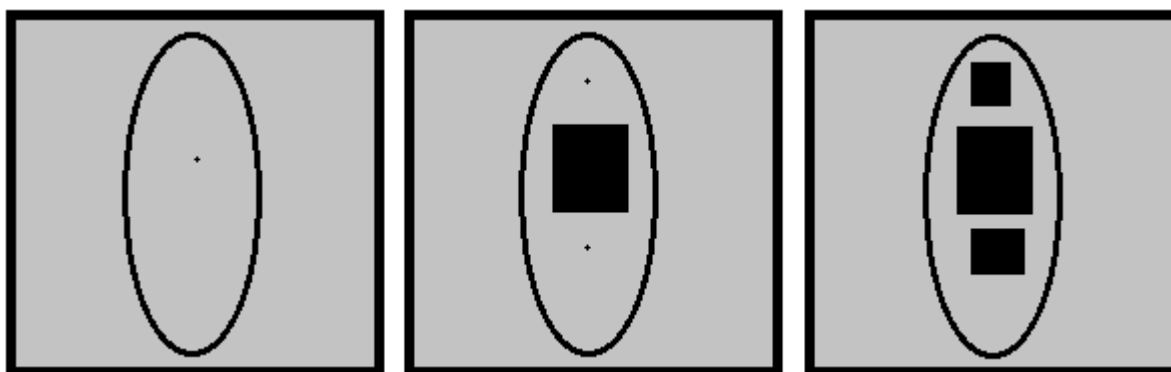


Obrázek 6: Vícedimenzionální představa: funkce Level-Set.

3.5 Region growing

Pixely s podobnou amplitudou, které leží vedle sebe jsou seskupovány k sobě a tím vytvoří segmentovanou oblast. V obraze se začíná pracovat s malými body a postupně se připojují další okolní pixely, které mají podobné vlastnosti. Jedna z těchto vlastností je úroveň šedé hodnoty pixelu. Proces segmentace je neustále opakován do té doby, než v obraze vzniknou oblasti, které se dají spojit.

U této segmentace se často setkáme s problémem, kde začít. Vybrat počáteční body, ze kterých se bude v obraze dále pokračovat v algoritmu, je v některých případech velmi složité. S dalším problémem se setkáme při spojování jednotlivých oblastí. Spojování oblastí závisí především na jasu, barvě a podobně. Jeden z posledních častých problémů metody spojování oblastí je i nastavení ukončujícího parametru této techniky. Na obrázku 7 je znázorněn průběh této metody.

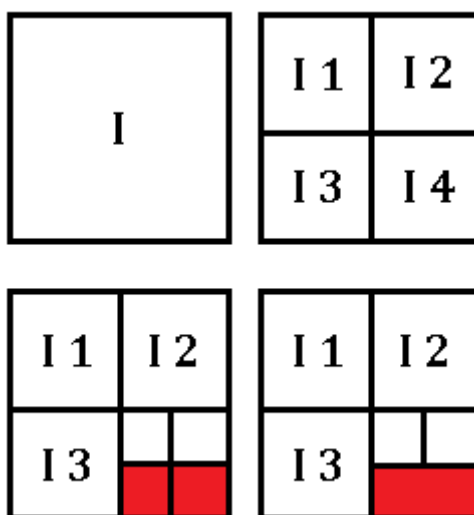


Obrázek 7: Průběh Region Growing: zleva doprava.

3.6 Split and merge

V této technice segmentace obrazu se začíná s jednou čtvercovou oblastí a tou je celý obraz. Poté je obraz dělen na čtyři kvadranty. Pokud jsou některé sousední čtverce homogenní, pak jsou ve výsledku sloučeny. Proces dělení pokračuje až do doby, kdy již není možno z hlediska homogenity spojit některé z vytvořených oblastí. Na obrázku 8 je ukázáno rozdělování do oblastí a červeně jsou vyznačena ty oblasti, které mají podobné vlastnosti, a proto jsou následně spojeny v jednu oblast.

Oblasti dělíme modální metodou nebo metodou diskriminantu, podrobnější informace naleznete v [5].



Obrázek 8: Dělení obrazu na jednotlivé kvadranty.

3.7 Prahování

Prahování je jedna z nejpoužívanější a nejoblíbenější metodou. Její použití je buď samostatné nebo je součástí jiných složitějších metod pro segmentaci obrazu. Vyniká svou jednoduchostí na implementaci a její časová složitost není vysoká. Mimo jiné se používá také při předzpracování obrazu pro již výše zmíněnou detekci hran v obraze nebo pro rozpoznávání znaků a podobně.

Myšlenka prahování je založena na tom, že objekty a pozadí mají rozdílnou úroveň intenzity jasu. V obraze je definovaný práh a každý pixel, který má menší hodnotu než definovaný práh, je určen jako pixel patřící do pozadí. Ostatní pixely jsou považovány za objektové pixely, čili pixely, které náleží přímo danému objektu. Po jediném průchodu prahování v obraze, dostaneme výsledek.

3.7.1 Předzpracování obrazu

Je hned několik způsobů, jak můžeme pomocí prahování obraz předzpracovat:

- Minimální filtr
- Morfologický tophat
- Procentní prahování
- Poloprahování
- Adaptivní prahování
- Vícetupňové prahování
- Hysteresní prahování

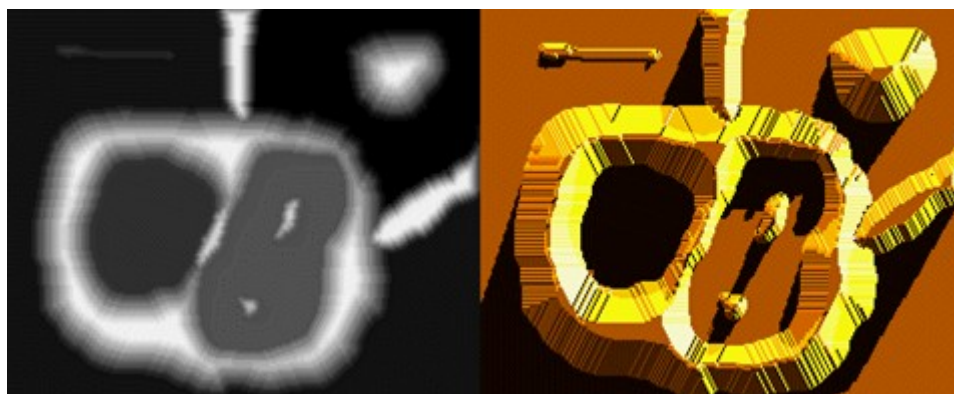
3.7.2 Automatické určování prahu

Existuje technika, která automaticky hledá vhodný práh. Tato technika je založena na analýze histogramu, ve kterém práh automaticky nachází.

Důležitým faktorem je histogram, ve kterém chceme nalézt práh a taktéž záleží na tom, jestli obsahuje jeden nebo více dominantních vrcholů.

4 Watershed

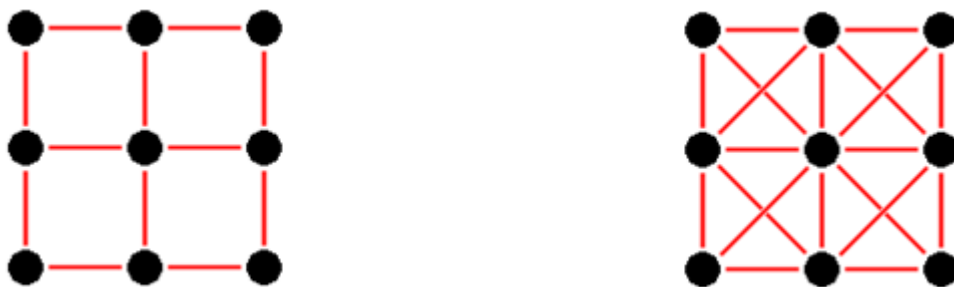
Segmentace obrazu povodím vychází z geografie. Na obraz se díváme jako na krajinu nebo topografický reliéf, který je postupně zaplavován vodou. Vodou může být zaplaven ze spod nebo padajícími kapkami vody shora dolů. Krajinu můžeme taktéž ponořit například do jezera, s propíchnutými otvory v lokálních minimech. Většinou se začne vodou zaplavovat terén od lokálních minim a v místech, kde se setkají povodí, se vytvoří vodní předěl neboli vodní hráz. Když hladina vody dosáhne nejvyššího vrcholu v krajině, proces je zastaven. Krajina je tedy rozdělena jednotlivými hrázemi na jednotlivé regiony, jednoduše na povodí [4]. Na obrázku 9 je vidět, jak se metoda Watershed dívá na obraz.



Obrázek 9: Obraz v šedé stupnici vlevo, je reprezentován jako topografický reliéf vpravo [4].

4.1 Důležité pojmy

Jelikož se díváme na obraz v šedé stupnici jako na topografický reliéf, je výška pixelu rovna hodnotě jasů pixelu. Jas nabývá hodnoty od 0 do 255. Dále pak zavádíme pojem sused. Jeden pixel v obraze může mít 4 nebo 8 okolních susedů. To jakým způsobem se nahlíží na okolní susedy, je zobrazeno na obrázku 10. U obrázku 10 vlevo, je vidět, že se středový pixel dostane maximálně na suseda nahoře, dole, vlevo a vpravo. Vpravo na obrázku 10 je pak vidět spojitost 8, kde se pixel, který je umístěn ve středu dostane na susedy co se týče šikmých směrů.

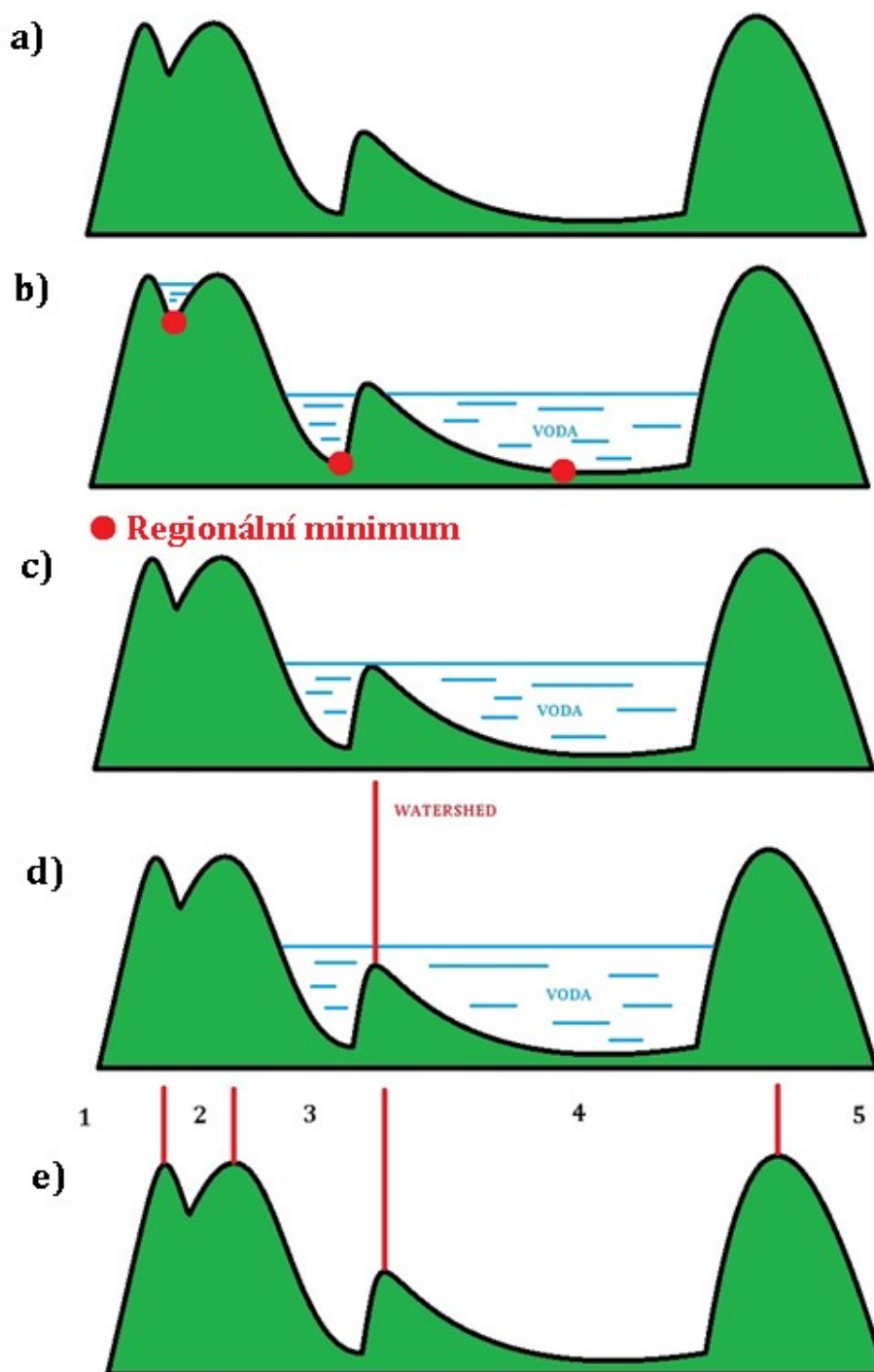


Obrázek 10: Nahlížení na okolní sousedy: vlevo spojitost 4, vpravo spojitost 8 [3].

Následuje pojem singulární minimum. Singulární minimum je místo v obraze, jehož sousedi jsou vždy vyšší. Poté následuje pojem minimální plošina a non-minimální plošina. Minimální plošina je plošina, ze které je možno sestoupit. Non-minimální plošina je plošina, ze které naopak sestoupit nemůžeme. Společně nám minimální a non-minimální plošina tvoří regionální minima v obrázku, ze kterých začíná proudit voda do obrazu [3].

4.2 Záplava

Tuto myšlenku zavedl v roce 1979 pan S. Beucher a C. Lantuéjoul [1]. Nejdříve si obraz představíme jako terén na obrázku 11 a). V terénu se určí regionální minima- to jsou nejnižší body v terénu, ze kterých se oblast začne zaplavovat vodou na obrázku 11 b). V průběhu zaplávání terénu vodou, dojde k setkání dvou povodí, tak jak je zobrazeno na obrázku 11 c). V místě, kde se tyto povodí setkaly, je postavena takzvaná vodní hráz neboli vodní předěl, zobrazen na obrázku 11 d). Jakmile je terén kompletně zaplaven vodou a jsou postaveny všechny hráze, dochází k rozdělení terénu na jednotlivé oblasti 1 až 5, zachycené na obrázku 11 e).



Obrázek 11: Průběh záplavy: a) terén, b) minima v terénu, c) setkání povodí, d) watershed hráz, e) finální rozdělení do oblastí.

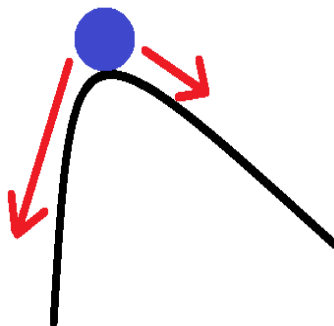
4.2.1 Vincent-Soille algoritmus

Tento algoritmus se vyznačuje svojí rychlostí. Je to především udáno tím, že segmentovaný obraz je doslova přesegmentovaný malými segmenty. Tento algoritmus má dva kroky. Prvním krokem je setřídění pixelů v závislosti na zvyšující se šedé úrovni jasů pixelů. Druhým krokem je zaplavovací proces, který začíná od minim v obraze. Implementace využívá *FIFO* (*first-in-first-out*) datovou strukturu pro uchování pixelů p [6].

Algoritmus přiřazuje odlišný label takzvaný štítek $lab[]$ každému minimum v obraze a jeho přidruženému povodí, které bylo zaplavováno s využitím breadth-first algoritmu [6]. V zaplavovacím procesu je všem sousedům s šedým levelem h dán label $MASK$. Dále sousedi, kteří oštitkovali sousedy z předchozí iterace jsou vloženi do fronty. Pokud má pixel dvě nebo více odlišných povodí, je označen za *watershed* souseda a je mu dán label $WSHED$. Když může být dosaženo pixelu pouze pomocí souseda, který má stejný label, soused je označen odpovídajícím povodím. Pixely, které mají na konci stále label $MASK$, patří k nastavení nového minima levelu h , jehož připojené komponenty dostanou nový label. Čas složitosti tohoto algoritmu je lineární s počtem pixelů ve vstupním obraze. Pseudokód tohoto algoritmu je k dostání v [6].

4.3 Déšť

Kromě zaplavovací techniky zespod, existuje ještě technika padajících kapek. Jde v podstatě o déšť, který padá shora na topografický relief. Jelikož o vodě víme, že si hledá cestu nejmenšího odporu, to znamená, že voda se bude snažit stékat vždy nejstrmější cestou. V nejstrmějších místech se pak nachází regionální minima tak, jak bylo vysvětleno u záplavy [3]. Na obrázku 12 je zobrazena kapka na topografickém reliéfu, která steče pravděpodobně na levou stranu, jelikož zde je plocha nejstrmější a nalezne tam regionální minimum.



Obrázek 12: Padající kapka si zvolí cestu nejmenšího odporu.

4.4 Meyerův algoritmus

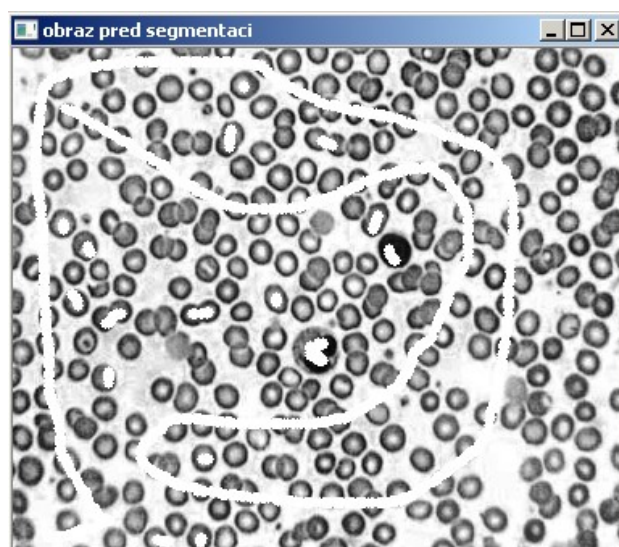
Jedním z nejstarších algoritmů stavicích na povodí představil na počátku 90. let pan F. Meyer. Algoritmus pracuje s šedou stupnicí obrazu. Zaplavovací process probíhá na přechodu snímku, to znamená, že povodí vznikají podél okrajů obrazu. Za normálních okolností tento algoritmus povede k nadměrné segmentaci obrazu, zejména pro lékařské obrázky, jako je například CT. Proto je lepší, mít obraz již předzpracovaný některou z metod pro předzpracování obrazu nebo musíme sloučit některé regiony na základě jejich podobnosti [2].

4.4.1 Princip

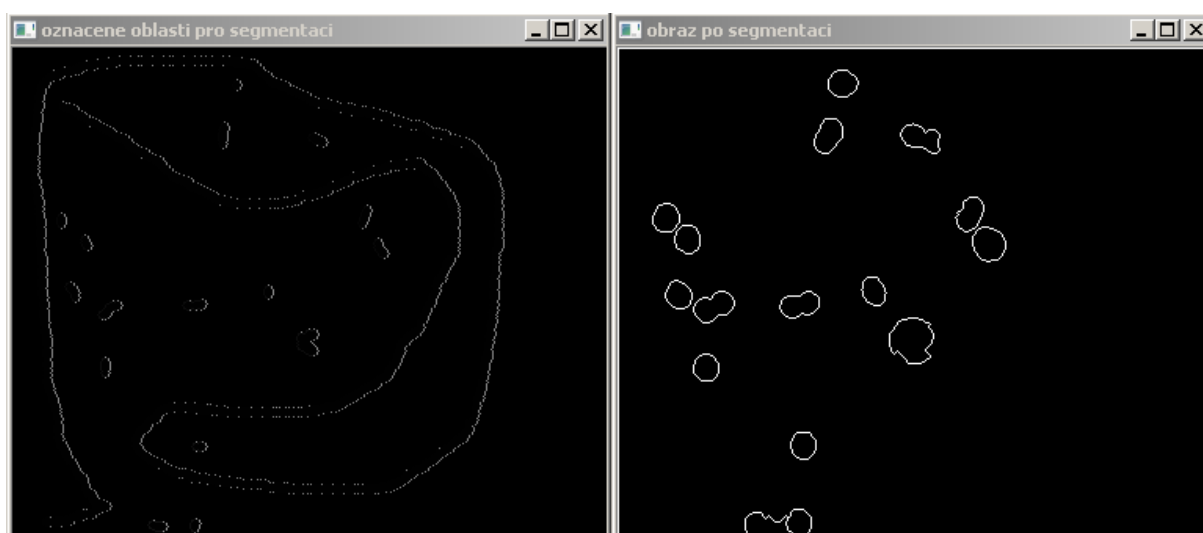
Na začátku je sada pixelů, takzvaných markerů, ze kterých se začíná zaplavovat. Každý marker má jiný label neboli štítek. Sousední pixely, které jsou z označených oblastí jsou vloženy do prioritní fronty s odpovídajícími úrovněmi šedi jednotlivých pixelů v obraze. Pixel, který má nejvyšší úrovnívou prioritu, je vyjmut z fronty. Pokud se stane, že sousedé pixelů vytěžených z fronty, které byly označené a mají stejné označení, je pixel označen jejich označením a všechny dosud neoznačené sousedi, kteří dosud nejsou v prioritní frontě jsou do fronty vloženy. Krok ve kterém se začíná vybírat pixel podle nejvyšší úrovníové priority se opakuje do té doby, dokud není fronta prázdná [2].

4.4.2 Ukázka

Na obrázku 13 je zobrazen vstupní obrázek buňek. Na tomto obrázku jsem provedl vyznačení těch objektů, které chci od sebe oddělit. Jak je vidno, označil jsem některé buňky a potom jsem dlouhou čarou zobrazil tu část obrazu, kterou chci od buňek oddělit. V mém případě jsem dlouhou čaru bral jako pozadí, které mě momentálně nezajímalo a zajímaly mne jen konkrétní buňky. Na obrázku 14 vlevo jsou zobrazeny oblasti, které jsem vybral pro začátek segmentace a vpravo je již výsledný obraz.



Obrázek 13: Obrázek buňek, s vyznačenými počátečními body pro začátek segmentace.



Obrázek 14: Vlevo označené oblasti pro začátek segmentace a vpravo již výsledný obraz s uzavřenými segmenty.

5 Implementace

5.1 Cíl

Cílem této implementace je implementovat algoritmus pomocí metody Watershed podle pseudokódu algoritmu od Vincenta a Soilleho [6]. Po provedení algoritmu bude výsledkem obraz s vyznačenými jednotlivými povodími, spočítání jednotlivých vytvořených povodí a v nesposlední řadě také výsledný čas segmentace. Implementovanou segmentaci použiji na více snímcích, s rozdílnou šířkou a výškou obrazu a rozdílným filtrem pro porovnání.

5.2 Implementace projektu Vincent Soille Algoritmus

Vincent Soille algoritmus jsem implementoval v prostředí Microsoft Visual Studio 2010 Ultimate, na které máme v rámci školy jakožto její studenti studentskou licenci pro nekomerční využití. K implementaci jsem dále zvolil jazyk C++ a využíval jsem knihovnu openCV a její funkce. Více o openCV, jeho funkcích a využití zde [9].

5.2.1 Obsah projektu

Součástí projektu jsou:

- Hlavičkový soubor *stdafx.h* sloužící ke kompilaci, tento hlavičkový soubor musí být vnořen do každého nového souboru s příponou *.cpp*.
- Inicializační soubor *stdafx.cpp*, odkazující na hlavičkový soubor *stdafx.h*.
- Hlavní soubor s výslednou implementací a funkcemi pro spuštění, vykreslení, počítání *VincentSoilleAlg.cpp*.
- Součástí projektu jsou také obrázky pro test segmentace.

Obsah *VincentSoilleAlg.cpp*:

- Potřebné hlavičkové soubory pro přístup k jednotlivým funkcím jazyka C++.
- Inicializační proměnné, které jsou definovány hned na začátku souboru.
- Jednotlivá makra pro přístup k pixelům v obrázku.
- Proměnné typu *IplImage*, reprezentující obrázky.

- Proměnné typu *CvMat* reprezentující matice obrázků.
- Pomocné proměnné typu *int*, *CvPoint*.
- Multimapa *Mmulti*, která slouží pro uchování jasů pixelů a jeho souřadnic.
- FIFO (first in first out) fronta *queue<CvPoint> fifo*.
- Funkce *pocetSegmentovanychPixelu()*, která slouží ke spočítání jednotlivých pixelů, které prošly algoritmem.
- Funkce *pocetNalezenychOblasti()*, která slouží k zobrazení počtu nalezených watershed oblastí.
- List *list<CvPoint> GetConn(CvPoint pocatecni_bod)*, který souží při prohledávání okolních sousedů. Pracuje se spojitostí 4. Díváme se tedy na okolní pixely doleva, doprava, nahoru, dolů.
- Funkce *kresli(CvMat *WshedMapa, IplImage *imI)*, která slouží pro vykreslení výsledné mřížky watershed pixelů do obrazu *imI* a následnému zobrazení na obrazovku a uložení obrázku pomocí funkce *cvSaveImage()*.
- *CvMat *VincentSoilleAlgoritmus(IplImage *imO)*, zde je obsažen celý algoritmus podle Vincent Soilleho pseudokódu a obsahuje také blur filtr *CvSmooth(imI, imI, CV_BLUR, 19, 19)*, který slouží ke zlepšení výsledku segmentace. Bez něj by byl výsledný obraz přesegmentovaný.
- Funkce *Main()*, hlavní spouštěcí funkce.

Kompletní projekt této segmentace je uložen na přiloženém CD disku v adresáři /Projekt.

5.2.2 Průběh segmentace

Na začátku segmentace dojde k inicializaci matic obrázků. Poté následuje cyklus *for*, který provádíme pro všechny hodnoty jasů neboli výšky pixelů od jasů minimálního až po jas maximální. V tomto případě jde o hodnoty 0 (černá) až 255 (bílá) jelikož pracujeme s šedou stupnicovou škálou.

Nejdříve označíme pixely maskou, poté se díváme na okolní sousedy a přiřazujeme jim patřičný label neboli štítek. Pokud má takovýto pixel hodnotu jasů větší než nula nebo je to přímo watershed pixel nastavíme tento pixel v mapě distanců na hodnotu 1 a pixel vložíme do fronty. Následuje nastavení aktuálního distance neboli vzdálenosti na hodnotu 1 a vložení fiktivního pixelu na konec.

Dále máme cyklus *while*, který vybírá pixely předtím vložené a pokud narazí na pixel fiktivní nebo je fronta vyprázdněna, končí, jinak vloží do fronty fiktivní pixel, zvýší aktuální distanc neboli

vzdálenost a zase ho vybere.

Následuje složitější cyklus *for*, ve kterém po projití všech okolních sousedů ukládáme hodnoty watershedu. Zde řešíme také plateau problém- pokud je stále nějaký pixel v label mapě označen maskou, a zároveň má v mapě distanců hodnotu 0, nastaví se mu distanc na distanc aktuální + 1 a je vložen do fronty.

Poslední cyklus *for* je cyklus, ve kterém detekujeme nové minima v matici obrazu. Pokud má stále hodnotu jasu nastavenou na masku, je mu dán nový štítek neboli label a je vložen do fronty. Potom dokud není fronta prázdná, vybíráme pixely z fronty a díváme se do jeho okolí. Pokud má pixel jas nastaven na masku, pixel vložíme zpět do fronty a jeho štítek nastavíme na štítek aktuální.

5.3 Výsledky

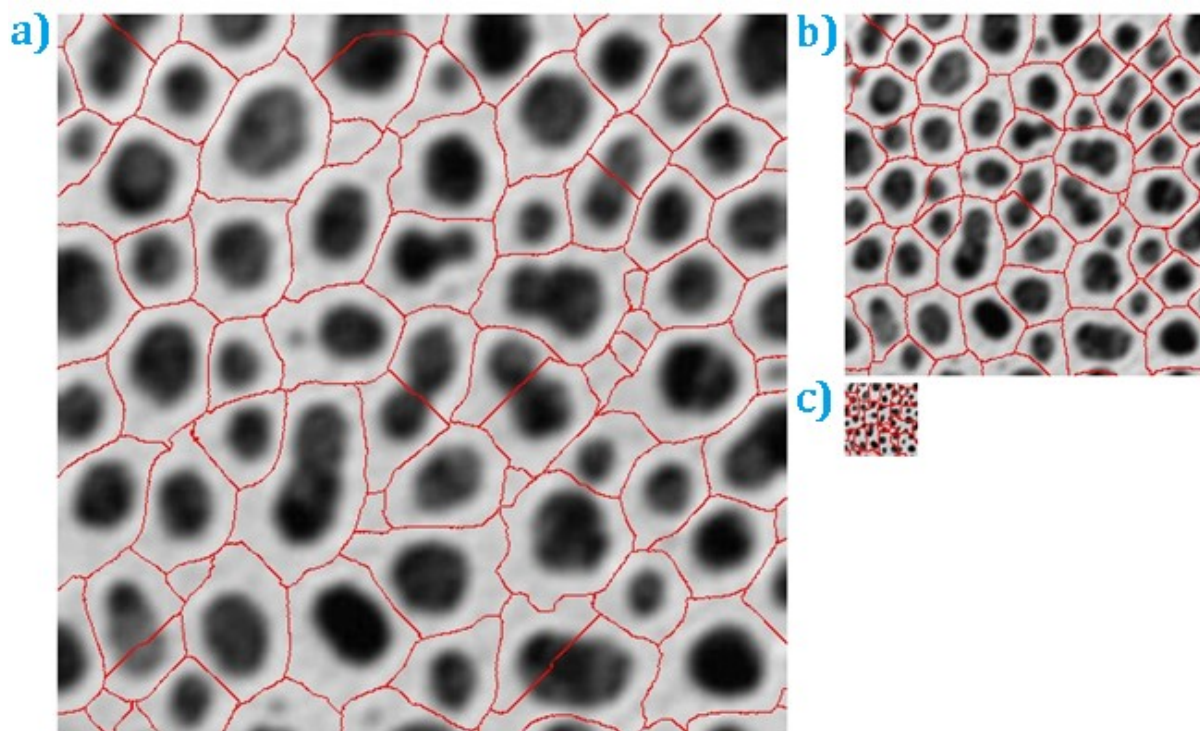
Vstupem je načítaný obrázek *imI*, obrázek má koncovku vždy *.jpg* a výstupem je matice watershed pixelů zobrazená právě na vstupním obrázku. Výsledky segmentace, zkoumání počtů segmentů a rychlosti segmentace na různých velikostech vstupních obrázcích zachycují následující testy. Test 1 A) byl prováděn na lékařském snímku s buňkami a byl proveden s využitím filtru. Test 1 B) byl prováděn taktéž na tom samém snímku, jako v Testu 1 A) s tím rozdílem, že při segmentaci nebyl použit žádný filtr. V Testu 2 jsem provedl segmentaci nad snímkem krajiny. Jednou s použitím filtru a podruhé bez použití filtru. Výsledky jsou zaznamenány v tabulkách 1, 2 a 3. V každé tabulce je i odkaz na konkrétní obrázek po segmentaci.

5.3.1 Test 1 A)

Vstupem pro tento test byl lékařský obrázek buněk v různých velikostech. Výstup tohoto testu je zobrazen na obrázku 15. Následující tabulka 1 nese informace o provedeném testu segmentace.

Výsledky segmentace s využitím filtru				
Rozměr v pixelech	Počet pixelů	Počet oblastí	Čas v milisekundách	Odkaz na obrázek
50 x 50	2 500	33	232,021	15 c)
250 x 248	62 000	63	5 684,01	15 b)
500 x 496	248 000	87	23 478	15 a)

Tabulka 1: Tabulka testu 1 A).



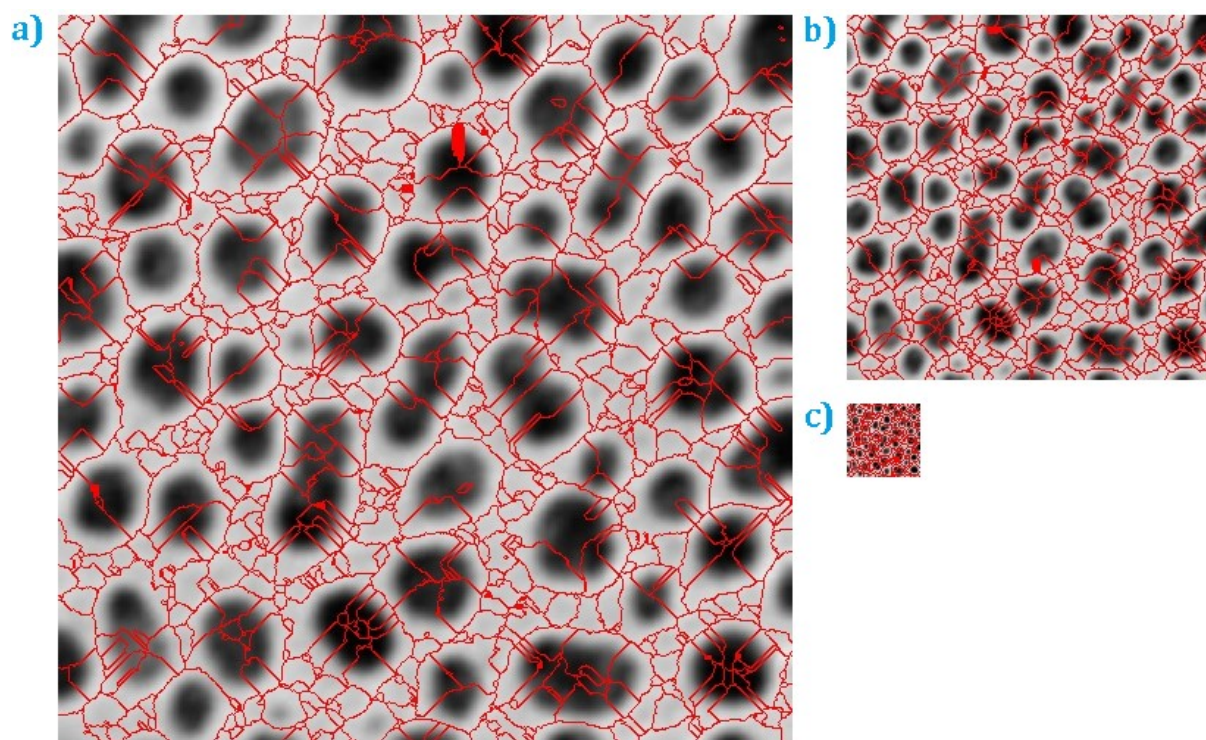
Obrázek 15: Zmenšený výstup Testu 1 A): a) 500 x 496, b) 250 x 248, c) 50 x 50.

5.3.2 Test 1 B)

Vstupem pro tento test byl lékařský obrázek buňek v různých velikostech. Výstup tohoto testu je zobrazen na obrázku 16. Následující tabulka 2 nese informace o provedeném testu segmentace.

Výsledky segmentace bez využití filtru				
Rozměr v pixelech	Počet pixelů	Počet oblastí	Čas v milisekundách	Odkaz na obrázek
50 x 50	2 500	161	254,291	16 c)
250 x 248	62 000	541	5 911,52	16 b)
500 x 496	248 000	861	23 756,2	16 a)

Tabulka 2: Tabulka Testu 1 B).



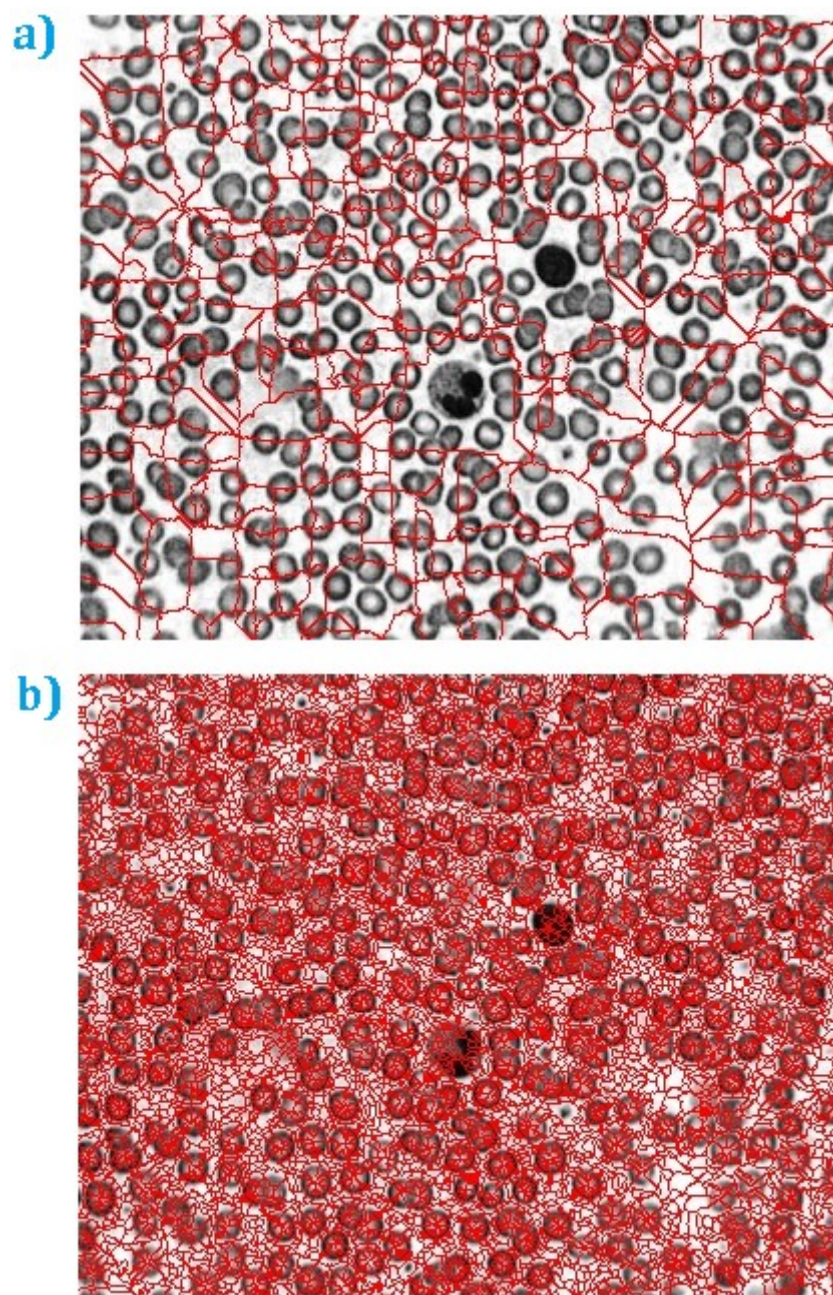
Obrázek 16: Zmenšený výstup Testu 1 B): a) 500 x 496, b) 250 x 248, c) 50 x 50.

5.3.3 Test 2

Vstupem pro tento test byl jiný lékařský obrázek o velikosti 379 krát 312 pixelů. Výstup tohoto testu je zobrazen na obrázku 17. Následující tabulka 3 nese informace o provedeném testu segmentace.

Výsledky segmentace s využitím filtru				
Rozměr v pixelech	Počet pixelů	Počet oblastí	Čas v milisekundách	Odkaz na obrázek
379 x 312	118 248	346	11 461,5	17 a)
Výsledky segmentace bez využití filtru				
379 x 312	118 248	6 701	10 665,7	17 b)

Tabulka 3: Tabulka Testu 2.



Obrázek 17: Výstup Testu 2: a) s filtrem, b) bez filtru

6 Závěr

Hlavním cílem této bakalářské práce bylo seznámení se s technikami segmentace obrazu, vybrat si jednu segmentaci obrazu a implementovat ji s pomocí techniky Watershed. Toto téma je tak rozsáhlé, že není možno jej celé objasnit v jedné bakalářské práci a proto se tímto tématem lze zabývat i v rámci diplomové práce a výše.

Pro implementaci jsem si vybral algoritmus Vincent Soille. Tento algoritmus se vyznačuje především svou rychlostí a jeho použití může být především v medicínské oblasti. Tato bakalářská práce mi nejen rozšířila obzor v oblasti zpracování obrazu, ale naučil jsem se pracovat s knihovnou openCV, která značně ulehčuje práci pro segmentaci obrazu, a obsahuje některé velice zajímavé funkce jako jsou různé filtry, které jdou použít na obraz. Dále jsem zjistil, že pokud při testování na snímcích z obrázku 16 nepoužiji žádný filtr, který rozmazává obraz, je segmentace mnohdy pomalejší o desítky až stovky milisekund, což je udáno především formátem daného obrázku. Počet segmentů vzrostl někdy až o pětinašobek u nejmenšího segmentovaného obrázku 50 na 50 pixelů a až o devítinašobek u obrázků větších- konkrétně u obrázku o velikosti 500 na 496 pixelů. Na obrázek 15 c) a 16 c) bychom již potřebovali lupu nebo větší přiblížení, kdybychom chtěli vidět výsledek segmentace detailně. Toto pro lidi pracující v medicíně není problém, jelikož využívají kvalitních mikroskopů, a tak jsou schopni vidět mnohonásobně menší objekty, čili by s tímto výsledkem segmentace byli schopni pracovat.

V Testu 2 na straně 24 jsem vyzkoušel segmentaci na jiném lékařském obrázku, kde se nachází více shluků buněk. V tomto testu se právě projevila rychlost samotné segmentace bez využití filtru a byla dokonce o 795,8 milisekund rychlejší a počet segmentů vzrostl oproti použití s filtrem o 19-ti násobek. Pokud bychom chtěli využít tento algoritmus na jiné snímky, jakožto například na snímky krajiny, museli bychom tento algoritmus spustit na úrovni gradientu obrazu a jestliže chceme co nejkvalitnější výsledek segmentace- co se přehlednosti týče, musíme vybírat mezi filtry ten správný, jelikož ne každý filtr nám rozostří obraz před zpracováním obrazu tak, jak bychom chtěli.

Literatura

1. BEUCHER, S. a LANTUÉJOUL, Ch. *Use of watersheds in contour detection. In International workshop on image processing, real-time edge and motion detection*, 1979. [Online]. [Citace: 2013-04-15]. Dostupné z: <http://cmm.ensmp.fr/~beucher/publi/watershed.pdf>
2. *Watershed (image processing) - Wikipedia, the free encyclopedia*. [Online]. [Citace: 2013-04-01]. Dostupné z: http://en.wikipedia.org/wiki/Watershed_%28image_processing%29
3. GOLODETZ, S. *Watersheds and Waterfalls. Overload Journal #83 - Feb 2008 + Programming Topics + Design of applications and programs*, (2008). [Online]. [Citace: 2013-04-05]. Dostupné z: <http://accu.org/index.php/journals/1469>
4. SBEUCHER, S. *Image Segmentation and Mathematical Morphology*. [Online]. [Citace: 2013-04-05]. Dostupné z: <http://cmm.ensmp.fr/~beucher/wtshed.html>
5. SOJKA, E. *Digitální zpracování a analýza obrazů*, 1. vydání, Ostrava: VŠB - Technická univerzita, 2000. ISBN 80-7078-746-5.
6. ROERDINK, J.B.T.M a MEIJSTER, A. *The Watershed Transform*, (2001). [Online]. [Citace: 2013-04-05]. Dostupné z: <http://www.cs.rug.nl/~roe/publications/parwshed.pdf>
7. CHEN, J. a PAPPAS, T. *Adaptive perceptual color-texture image segmentation*. [Online]. [Citace: 2013-04-06]. Dostupné z: <http://spie.org/x8899.xml?pf=true&ArticleID=x8899>
8. ŠPANĚL, M. a BERAN, V. *Obrazové segmentační techniky*, VUT - BRNO. [Online]. [Citace: 2013-04-06]. Dostupné z: <http://www.fit.vutbr.cz/~spanel/segmentace/>
9. *OpenCV*. [Online]. [Citace: 2013-04-06]. Dostupné z: <http://opencv.org/>

Adresářová struktura přiloženého CD

/Projekt	Kompletní projekt psán v jazyce C++.
/Text	Kompletní bakalářská práce.
/VysledkyTestu	Zde jsou umístěny obrázky z výsledků testů.